CSA1618 DWDM-DE

EXPERIMENT-28

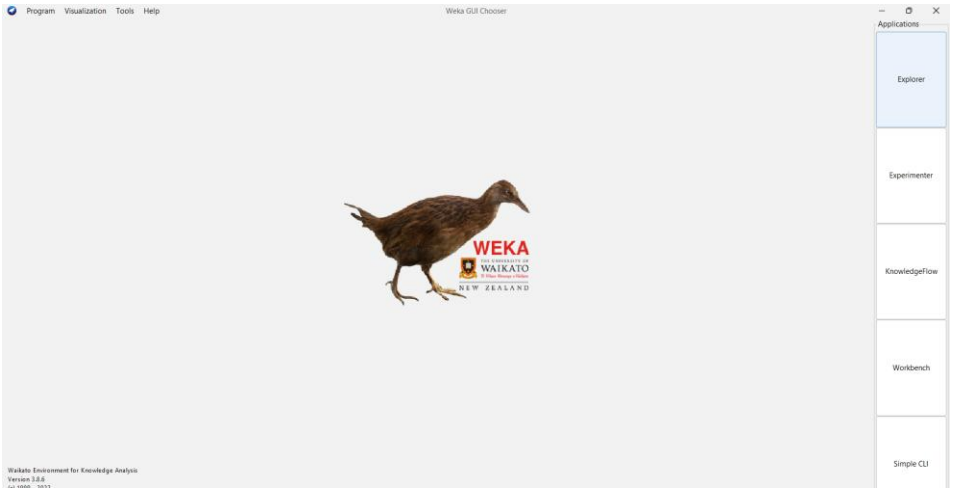
PREDICTION OF CATEGORICAL DATA USING SMO ALGORTIHM THROUGH WEKA

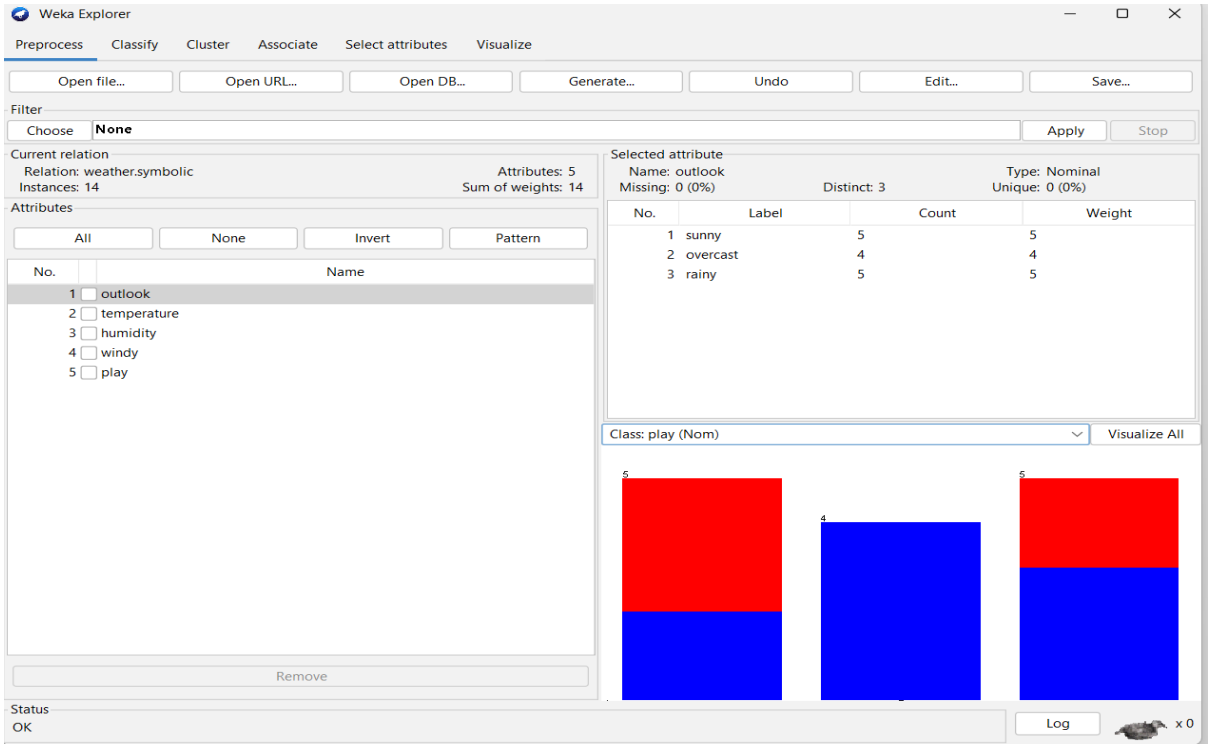
AIM:

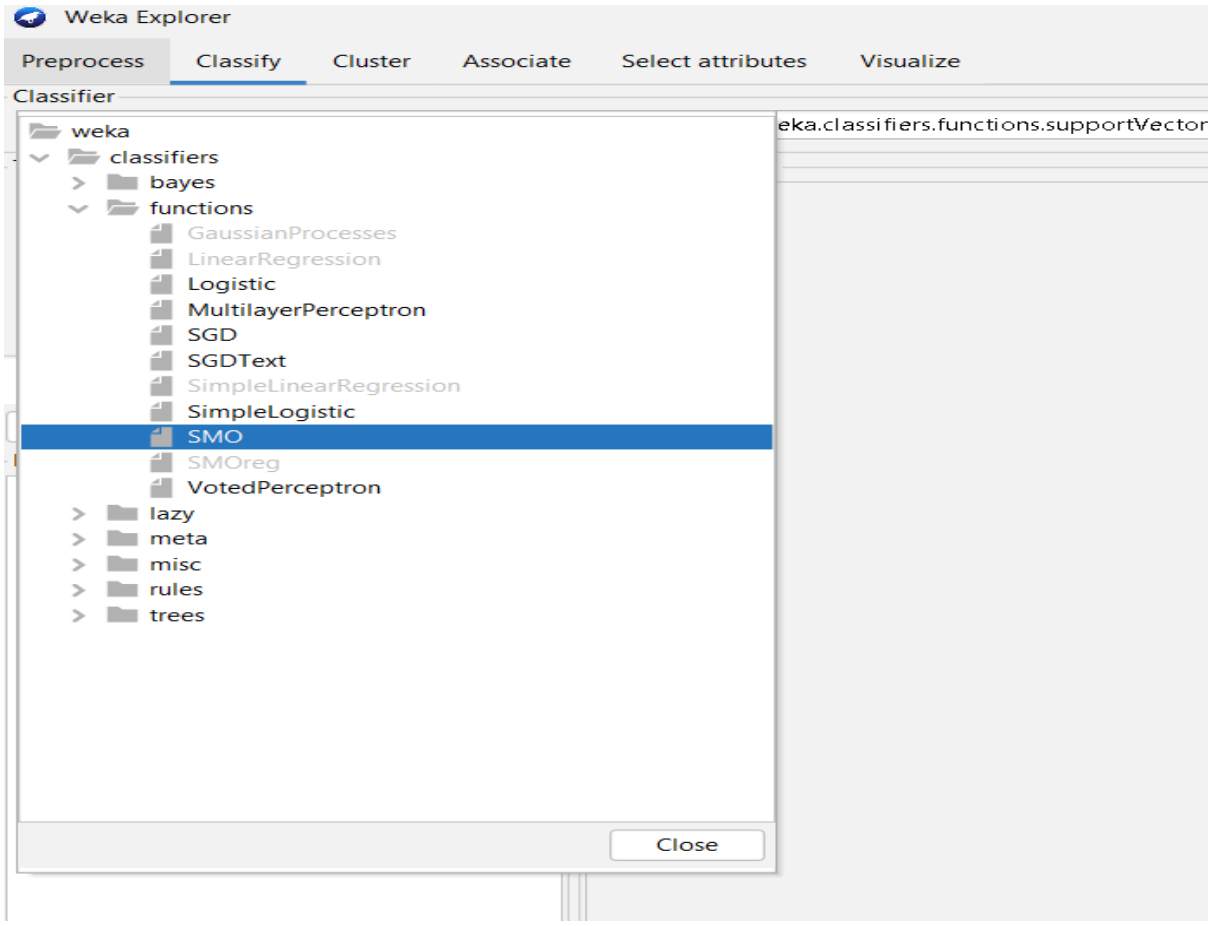
To create prediction of categorical data using SMO Algorithm through weka tool.

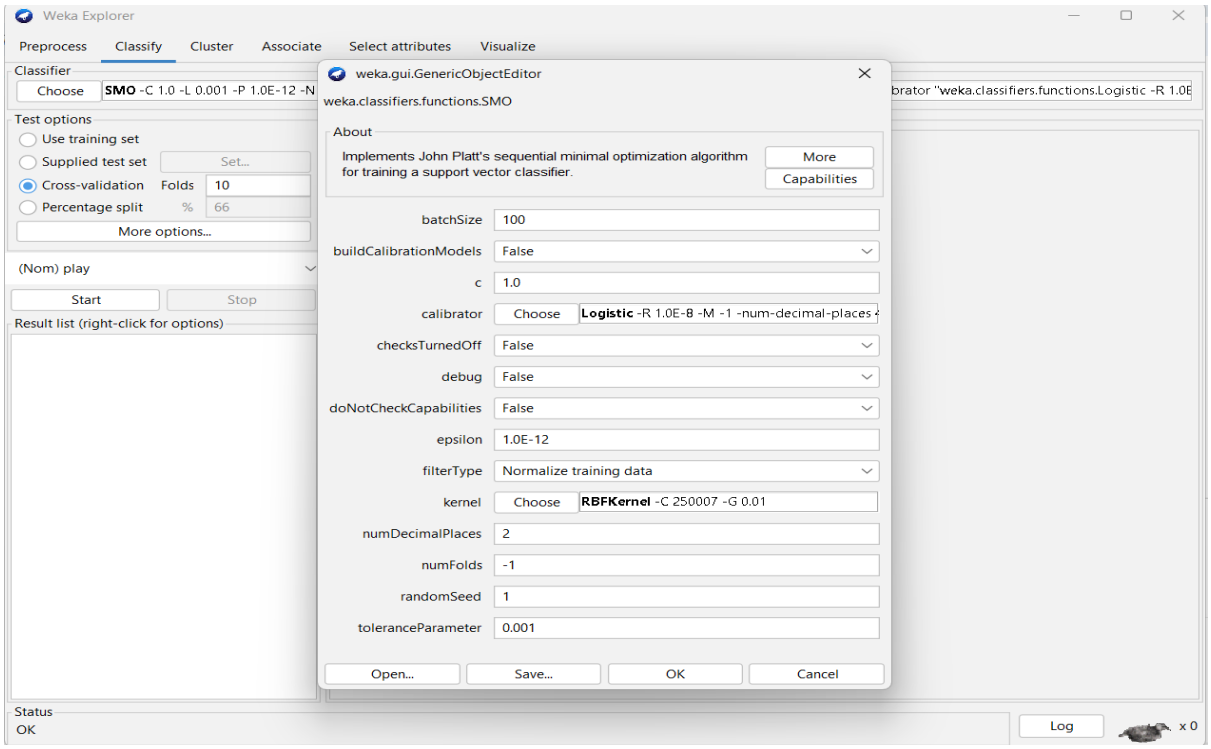
PROCEDURE:

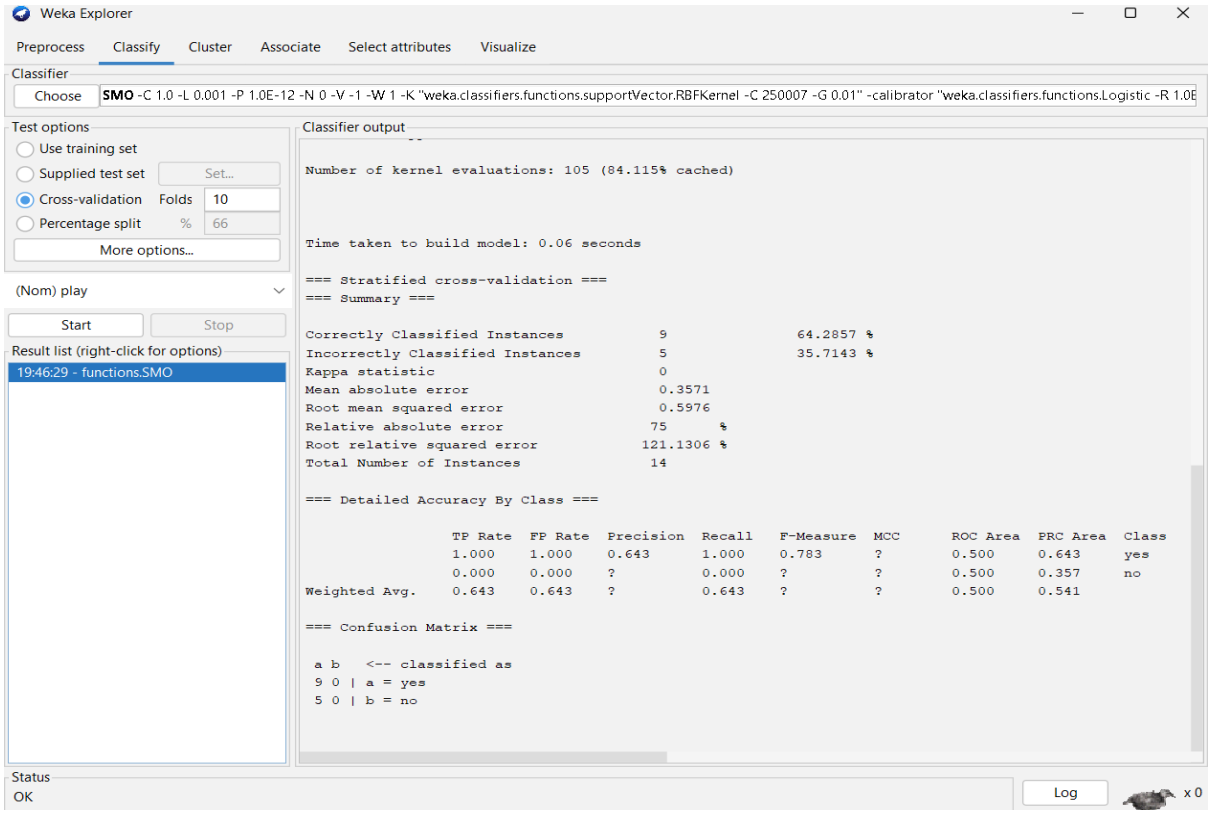
1. Download and install WEKA.
2. Open WEKA and Choose "Explorer" from the main menu.
3. Under Preprocess, Click on the open file button and select the dataset. Ensure that your dataset contains categorical (nominal) attributes.
4. Go to the Classify tab.
5. Click Choose → Expand the functions section → Select SMO (Sequential Minimal Optimization).
6. Click on SMO to open parameter settings : Kernel Type (K): Default is PolyKernel (Polynomial Kernel), change it to RBFKernel (Radial Basis Function, good for complex data), C value (C): Regularization parameter (default is 1.0), Epsilon (E): Controls the stopping criterion (default 1.0E-12).
7. Click OK to save settings.




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: weather.symbolic
Instances: 14
Attributes: 5
Sum of weights: 14
Attributes
All
None
Invert
Pattern
No.
Name
1 outlook
2 temperature
3 humidity
4 windy
5 play
Remove
Selected attribute
Name: outlook
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)
No.
Label
Count
Weight
1 sunny 5 5
2 overcast 4 4
3 rainy 5 5
Class: play (Nom)
Visualize All
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
eka.classifiers.functions.supportVector
weka
classifiers
bayes
functions
GaussianProcesses
LinearRegression
Logistic
MultilayerPerceptron
SGD
SGDText
SimpleLinearRegression
SimpleLogistic
SMO
SMOreg
VotedPerceptron
lazy
meta
misc
rules
trees
Close


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) play
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0
weka.gui.GenericObjectEditor
weka.classifiers.functions.SMO
About
Implements John Platt's sequential minimal optimization algorithm for training a support vector classifier.
More
Capabilities
batchSize
100
buildCalibrationModels
False
c
1.0
calibrator
Choose
Logistic -R 1.0E-8 -M -1 -num-decimal-places
checksTurnedOff
False
debug
False
doNotCheckCapabilities
False
epsilon
1.0E-12
filterType
Normalize training data
kernel
Choose
RBFKernel -C 250007 -G 0.01
numDecimalPlaces
2
numFolds
-1
randomSeed
1
toleranceParameter
0.001
Open...
Save...
OK
Cancel


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.RBFKernel -C 250007 -G 0.01" -calibrator "weka.classifiers.functions.Logistic -R 1.0E
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) play
Start
Stop
Result list (right-click for options)
19:46:29 - functions.SMO
Classifier output
Number of kernel evaluations: 105 (84.115% cached)
Time taken to build model: 0.06 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 9 64.2857 %
Incorrectly Classified Instances 5 35.7143 %
Kappa statistic 0
Mean absolute error 0.3571
Root mean squared error 0.5976
Relative absolute error 75 %
Root relative squared error 121.1306 %
Total Number of Instances 14
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
1.000 1.000 0.643 1.000 0.783 ? 0.500 0.643 yes
0.000 0.000 ? 0.000 ? ? 0.500 0.357 no
Weighted Avg. 0.643 0.643 ? 0.643 ? ? 0.500 0.541
=== Confusion Matrix ===
a b <-- classified as
9 0 | a = yes
5 0 | b = no
Status
OK
Log
x 0

OBSERVATION:

=== Classifier model (full training set) ===

SMO

Kernel used:

RBF Kernel: K(x,y) = exp(-0.01*(x-y)^2)

Classifier for classes: yes, no

BinarySMO

1 * <0 0 1 0 1 0 0 0 > * X]

\- 1 * <1 0 0 0 1 0 1 0 > * X]

\+ 1 * <1 0 0 0 1 0 0 1 > * X]

\+ 1 * <1 0 0 1 0 0 0 0 > * X]

\- 1 * <0 0 1 0 1 0 0 1 > * X]

\- 0.3383 * <0 0 1 0 0 1 1 1 > * X]

\+ 1 * <1 0 0 1 0 0 0 1 > * X]

\- 1 * <1 0 0 0 0 1 1 1 > * X]

\- 0.6617 * <0 1 0 0 1 0 0 0 > * X]

\- 1 * <0 1 0 1 0 0 0 1 > * X]

\+ 1 * <0 0 1 0 0 1 1 0 > * X]

\- 0.9805

Number of support vectors: 11

Number of kernel evaluations: 105 (84.115% cached)

Time taken to build model: 0.06 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	9	64.2857 %
Incorrectly Classified Instances	5	35.7143 %
Kappa statistic	0	
Mean absolute error	0.3571	
Root mean squared error	0.5976	
Relative absolute error	75 %	
Root relative squared error	121.1306 %	
Total Number of Instances	14	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	
	1.000	1.000	0.643	1.000	0.783	?	0.500	0.643	yes
	0.000	0.000	?	0.000	?	?	0.500	0.357	no
Wt Avg.	0.643	0.643	?	0.643	?	?	0.500	0.541	

=== Confusion Matrix ===

a	b	<-- classified as
9	0	a = yes
5	0	b = no

RESULT:

Thus, the observations and evaluations done on the dataset are analyzed. The implementation of Sequential Minimal Optimization (SMO) has been successfully visualized.